

RECEIVED
CENTRAL FAX CENTER

SEP 27 2006

REMARKS

Claims 1-26 have been presented for examination. In this Response, Applicant does not amend, cancel, or add any claims. Claims 1-26 remain in the Application. Reconsideration of the pending claims is respectfully requested in view of the following remarks.

I. Claims Rejected Under 35 U.S.C. § 102

Claims 1-5, 7-17, 19-24, and 26 stand rejected under 35 U.S.C. § 102(a) as being anticipated by U.S. Patent No. 6,938,254 issued to Mathur et al. ("Mathur"). Applicant respectfully traverses the rejection.

To anticipate a claim, the Examiner must show that a single reference teaches each of the elements of that claim. Among other elements, Claim 1 recites:

"A method of terminating an affected application program thread, comprising:
receiving an indication of a hardware error associated with an application program thread;
determining the application program thread to be in a user operation mode; and
terminating the application program."

Mathur discloses a method of controlling memory usage in a computer system having limited physical memory (Abstract). When the memory usage reaches a critical threshold, a user is prompted to select a currently executing application program to be terminated (col. 4, lines 56-59). Mathur does not disclose any indication of a hardware error in the above cited passage or elsewhere in the disclosure.

In the Final Office Action at page 8, the Examiner indicates that an error occurs when memory usage reaches the threshold or critical memory usage. The Examiner further indicates that "since this error affects memory which is hardware, then Mathur discloses a hardware error."

A hardware error, as generally known in the computer art, is an "error resulting from a malfunction of some physical component of the computer" (see, e.g., the definition provided by wordnet.princeton.edu). Critical memory usage is not an error caused by malfunction of a physical component of the computer. Critical memory usage may affect the use of memory, but is not related to hardware malfunction.

Applying the Examiner's analysis of the "hardware error," even a typical software error that affects memory allocation would be considered as a hardware error. For example, the

malloc subroutine in the C programming language's standard library performs dynamic memory allocation (see, e.g., "The C programming language" by Kernighan and Ritchie, Prentice-Hall, 1978). When the memory allocated by malloc exceeds the available memory size, an error occurs in the C program that calls the malloc subroutine. This malloc error is a software error because it is caused by unsuccessful execution of software code. However, applying the Examiner's analysis, a malloc error would be considered as a hardware error, because it affects the memory in the same way as the overused memory of Mathur. Thus, the above example shows that the Examiner's assertion is not consistent with the plain meaning of a hardware error as would be understood by one of ordinary skill in the art. Thus, the Examiner has not established that Mathur discloses "receiving an indication of a hardware error associated with an application program thread."

As previously mentioned, Mathur discloses that the user is prompted when the memory usage reaches a critical threshold. Mathur does not disclose that the user is prompted when a hardware error occurs. Mathur contemplates a computing environment in which application programs compete for available memory (col. 4, lines 12-14). Thus, in the context of Mathur, the critical threshold is reached when the application programs overuse their memory resources. As a result, one or more of the application programs are terminated to release a portion of the memory (Abstract). Thus, Mathur does not disclose the claimed "indication of a hardware error" associated with an application program thread which results in the termination the application program.

Analogous discussions apply to independent Claims 8, 13, and 20 and their dependent claims which incorporate the limitations thereof. Accordingly, reconsideration and withdrawal of the anticipation rejection of Claims 1-5, 7-17, 19-24 and 26 are requested.

II. Claims Rejected Under 35 U.S.C. § 103(a)

Claims 6, 18 and 25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable in view of Mathur over U.S. Patent No. 6,594,785 issued to Gilbertson et al. ("Gilbertson"). Applicant respectfully traverses the rejection.

To establish a *prima facie* case of obviousness, the relied upon references must teach or suggest every limitation of the claim such that the invention as a whole would have been obvious at the time the invention was made to one skilled in the art. Claims 6, 18 and 25 depend from

Claims 1, 13 and 20 and incorporate the limitations thereof. Thus, for at least the reasons mentioned above in regard to Claim 1, Mathur does not teach or suggest each of the elements of these dependent Claims.

Claim 6 modifies Claim 1 to include "receiving information of a poisoned data address associated with the hardware error." The Examiner relies on Gilbertson for disclosing the claimed "receiving information of a poisoned data address associated with the hardware error." However, Gilbertson does not cure the deficiency of Mathur, which fails to disclose "receiving an indication of a hardware error associated with an application program thread" recited in base Claim 1. Gilbertson discloses a multiprocessor system that isolates faults within a failing partition (e.g., a processor) and prevents the faults from creating a failure in a non-failing partition (col. 1, lines 46-50). However, Gilbertson does not teach or suggest the concept of "a hardware error associated with an application program thread." The Examiner has not identified and Applicant has been unable to discern any part of Gilbertson that discloses fault isolation based on the application program thread with which a hardware error is associated. Rather, Gilbertson isolates faults based on the physical partition with which a hardware error is associated. Thus, Mathur in view of Gilbertson does not teach or suggest each of the elements of Claim 1 and its dependent Claim 6.

Moreover, there is no motivation to combine Mathur with Gilbertson. Mathur discloses a method for reducing memory usage by terminating at least one application program. Thus, the method of Mathur handles software errors by terminating software programs. Gilbertson discloses fault handling by poisoning specific memory locations. Thus, the method of Gilbertson handles hardware errors by invalidating faulty hardware. The two references propose distinctly different solutions for solving distinctly different technical problems. Thus, a skilled person would not be motivated to combine the references. See MPEP § 2143.01 I. The Prior Art Must Suggest the Desirability of the Claimed Invention. Thus, the proposed combination is inapposite.

Accordingly, reconsideration and withdrawal of the obviousness rejection of Claims 6, 18 and 25 are requested.

SEP 27 2006

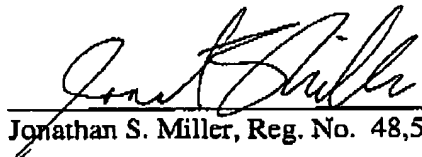
CONCLUSION

In view of the foregoing, it is believed that all claims are now in condition for allowance and such action is earnestly solicited at the earliest possible date. If there are any additional fees due in connection with the filing of this response, please charge those fees to our Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

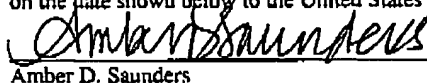
Dated: September 27, 2006


Jonathan S. Miller, Reg. No. 48,534

12400 Wilshire Blvd.
Seventh Floor
Los Angeles, California 90025
(310) 207-3800

CERTIFICATE OF FACSIMILE

I hereby certify that this correspondence is being transmitted via facsimile on the date shown below to the United States Patent and Trademark Office.

 9/27/06
Amber D. Saunders Date